LA-UR-01-1533
*Approved for public release;
distribution is unlimited.*

Title:

FULLY CONSISTENT, LINEAR DISCONTINUOUS DIFFUSION SYNTHETIC ACCELERATION ON 3D UNSTRUCTURED MESHES

Author(s): James S. Warsa, Todd A. Wareing and Jim E. Morel

Submitted to:

http://lib-www.lanl.gov/la-pubs/00818581.pdf

M&C 2001 Salt Lake City, Utah, USA, September 2001.

FULLY CONSISTENT, LINEAR DISCONTINUOUS DIFFUSION SYNTHETIC ACCELERATION ON 3D UNSTRUCTURED MESHES

James S. Warsa, Todd A. Wareing and Jim E. Morel
Los Alamos National Laboratory
Transport Methods Group, MS D409
Los Alamos, New Mexico 87545
warsa@lanl.gov, wareing@lanl.gov, jim@lanl.gov

Keywords: diffusion synthetic acceleration, discrete ordinates transport, discontinuous finite element method, multi–level preconditioning, Fourier analysis

ABSTRACT

We extend a multi–level preconditioned solution method for a linear discontinuous discretization of the P_1 equations in two–dimensional Cartesian geometry to three–dimensional, unstructured tetrahedral meshes. A diffusion synthetic acceleration (DSA) method based on these P_1 equations is applied to linear discontinuous S_N transport source iterations on tetrahedral meshes. It is a fully consistent method because the P_1 equations and the transport equation are both discretized with a linear discontinuous finite element basis. Fourier analyses and computational results show the DSA scheme is stable and very effective. We compare the fully consistent scheme to other "partially consistent" DSA methods.

1 INTRODUCTION

An effective DSA scheme is often necessary for the efficient solution of S_N transport source iteration. It is well known that the discretization of the diffusion equation used in a DSA scheme has to be "consistent" with the transport equation discretization in order for it to be effective and robust over a wide range of problems (Alcouffe, 1977, Larsen, 1982).

DSA for linear discontinuous finite element discretizations of the S_N equations requires a linear discontinuous finite element discretization of the P_1 equations. This can be done by using a linear finite element basis to represent the P_1 solution combined with a way to introduce discontinuities into the discretization. Because it is a consistent discretization the corresponding DSA scheme will be effective and relatively few source iterations will be needed. However, when solving large 3D problems on unstructured grids, the P_1 linear system can be quite large. Iterative solution of this system can converge slowly, while direct methods are impractical.

To address this difficulty, DSA schemes have been developed that are only partially consistent with the underlying discontinuous finite element discretizations of the transport equation (Wareing, 1991, Adams, 1992, Morel, 1993, Wareing, 2001). Such DSA schemes can be very efficient. But they can also be although less effective than a consistent method in just the kinds problems for which DSA is most needed. A partially consistent DSA scheme could even possibly cause the transport iterations to become unstable. In particular, we will show that a scheme previously found to be stable in 1D and 2D Cartesian geometries (Adams, 1992) is unstable on 3D unstructured meshes.

In contrast, our approach is to iteratively solve the linear discontinuous P_1 equations as efficiently as possible with a robust preconditioner. The two–level preconditioner presented by Warsa, et al. (2000) was shown to be very effective in two–dimensional Cartesian geometry. In this work we extend that preconditioned solution technique to 3D unstructured tetrahedral meshes. The method is then implemented as a fully consistent DSA scheme for a linear discontinuous S_N transport code (Wareing, 1996).

The remainder of the paper is organized as follows. In the next section we describe how we discretize and solve the three–dimensional, discontinuous P_1 equations with our two–level preconditioner. The third section describes a Fourier analysis that can predict the spectral radius of a DSA method on 3D tetrahedral meshes. In the fourth section we compare the Fourier analysis predictions to spectral radius measurements from an implementation code. A moderately sized example problem is used to measure the computational effort of the accelerated S_N solution methods. All the results compare the fully consistent method to other partially consistent DSA schemes. The paper concludes with a few summary remarks.

2 DISCONTINUOUS P_1 EQUATIONS ON TETRAHEDRAL MESHES

In this work, we derive a discretization of the P_1 equations to be used as a fully consistent DSA method for discontinuous finite element discretizations (DFEM) of the S_N transport equation.

In this section we describe the discontinuous discretization of the P_1 equations on tetrahedra. That is followed by a description of the two–level preconditioning approach using a continuous discretization of the diffusion equation. The section concludes with a brief discussion of the DSA algorithm.

2.1 The Discretized Equations

We start with the steady–state P_1 system of equations. In three–dimensional geometry they are

$$\nabla \cdot \mathbf{J}(\mathbf{r}) + \sigma_a(\mathbf{r})\,\Phi(\mathbf{r}) = Q_0(\mathbf{r}) \tag{1a}$$

$$\frac{1}{3}\nabla\Phi(\mathbf{r}) + \sigma_t(\mathbf{r})\,\mathbf{J}(\mathbf{r}) = \mathbf{Q}_1(\mathbf{r}). \tag{1b}$$

The usual particle transport notation is used: $\Phi(\mathbf{r})$ represents the scalar flux (zeroth moment of the angular flux), $\mathbf{J}(\mathbf{r})$ represents the current (first moment of the angular flux). The source terms $Q_0(\mathbf{r})$ and $\mathbf{Q}_1(\mathbf{r})$ are the zeroth and first angular moments of an inhomogeneous source, a material property. The first expression is the balance equation. The second is a vector equation we will refer to as the first moment equation(s). They will consistently be written in this order.

Boundary conditions for the P_1 equations are specified by separating the flow of particles through a surface into inwardly and outwardly directed flows using partial currents. The inwardly directed flow (partial current) of particles through a surface located at $\mathbf{r}_s$ can be expressed as

$$J^{in}(\mathbf{r}_s) = \int_{\left(\hat{n}\cdot\hat{\Omega}\right)<0} \left(\hat{n}\cdot\hat{\Omega}\right)\,\Psi(\mathbf{r}_s,\hat{\Omega}) = \frac{1}{4}\Phi(\mathbf{r}_s) - \frac{1}{2}\hat{n}\cdot\mathbf{J}(\mathbf{r}_s) \tag{2a}$$

Similarly, an outwardly directed flow is given by

$$J^{out}(\mathbf{r}_s) = \int_{\left(\hat{n}\cdot\hat{\Omega}\right)>0} \left(\hat{n}\cdot\hat{\Omega}\right)\,\Psi(\mathbf{r}_s,\hat{\Omega}) = \frac{1}{4}\Phi(\mathbf{r}_s) + \frac{1}{2}\hat{n}\cdot\mathbf{J}(\mathbf{r}_s) \tag{2b}$$

Assume no external angular flux of particles enters through the external boundary surface $\Gamma \subset \Omega$, that is, $\Psi(\mathbf{r_s},\hat{\Omega}) = 0$ for $\mathbf{r}_s \in \Gamma$ and $\left(\hat{n}\cdot\hat{\Omega}\right) < 0$. Then the vacuum Marshak boundary conditions relate the scalar flux and currents on the boundary through Eq. 2a by setting $J^{in}(\mathbf{r}_s) = 0$ for $\mathbf{r}_s \in \Gamma$. Reflection boundary conditions are specified by setting $J^{in}(\mathbf{r}_s) = J^{out}(\mathbf{r}_s)$.

We use a linear finite element basis on a tetrahedral cell $T_k \in \Omega$. A local ordering of the faces and vertices of a cell is established such that the face i is opposite the vertex i. The linear basis functions functions u_i and $\mathbf{w}_i$, $i = 1,\ldots,4$, are defined by the linear barycentric coordinates on a "master" tetrahedron. There is a unique mapping between the actual coordinates of the mesh vertices and the local barycentric coordinates. More on the use of barycentric coordinates can be found in (Zienkiewicz, 1994). Their use simplifies the derivation and is easily generalized to higher order elements (Wareing, 2001).

We can construct a DFEM for the P$_1$ equations as follows. The boundary of a tetrahedral cell k is ∂T_k, consisting of each of the four faces $\mathbf{a}_j$, $j = 1,4$, opposite the four respective vertices. With the trial functions u_h and $\mathbf{w}_h$, the discrete problem is to solve

$$\int_{\partial T_k} \left(\hat{n}\cdot\mathbf{J}_h^b\right) u_h\,dS - \int_{T_k} \mathbf{J}_h\cdot\nabla u_h\,dV + \sigma_{a,k}\int_{T_k} \Phi_h u_h\,dV = \int_{T_k} Q_0 u_h\,dV \tag{3a}$$

$$\frac{1}{3}\int_{\partial T_k} \Phi_h^b\left(\hat{n}\cdot\mathbf{w}_h\right)\,dS - \frac{1}{3}\int_{T_k} \Phi_h\left(\nabla\cdot\mathbf{w}_h\right)\,dV + \sigma_{t,k}\int_{T_k} \mathbf{J}_h\cdot\mathbf{w}_h\,dV = \int_{T_k} \mathbf{Q}_1\cdot\mathbf{w}_h\,dV, \tag{3b}$$

for Φ_h and $\mathbf{J}_h$, which are linear approximations, expanded in terms of the basis functions u_i and $\mathbf{w}_i$, to the scalar flux and current on the element T_k. Note that we have used the divergence theorem to integrate terms involving gradients. This is necessary for introducing the discontinuous approximation.

For our linear trial space there are four vector trial functions $\mathbf{w}_h$ and four scalar trial functions u_h. Equations 3 are written at each vertex of every cell and Eq. 3b is written separately for each of the x, y, and z components at the four cell vertices. The result is sixteen equations in sixteen unknowns for each cell in the mesh. Because the basis functions are unity at their respective vertices and are zero at the other three vertices, the unknowns on a cell are the values of flux and three current components at the four cell vertices, that is, $\Phi_i, i = 1,4$ and J_i^x, J_i^y, and $J_i^z, i = 1,4$. These values are discontinuous, that is, they are separately defined on each cell as the limiting values of $\Phi(\mathbf{r})$ and $\mathbf{J}(\mathbf{r})$ as $\mathbf{r}\rightarrow\mathbf{v}_i$ from within the cell.

The integrals over the faces of the tetrahedral cells, ∂T_k, contain the quantities Φ_h^b and $\mathbf{J}_h^b$, indicating that they are "boundary" terms. They can be uniquely defined in terms of the particle flows through the surface between two adjacent cells using discrete versions of the partial currents in Eqs. 2a and 2b.

The result of the integrations over the cell boundaries in Eqs. 3 involve quantities of the form $\mathbf{a}_j\Phi_i^b$ and $\left(\mathbf{a}_j\cdot\mathbf{J}_i^b\right)$ at vertex i. The "area vector" is defined as $\mathbf{a}_j = \hat{n}_j a_j$ for a face j whose area is a_j with outward normal $\hat{n}_j$. We can "upwind" and account for boundary conditions or internal faces using discrete versions of the partial currents as follows. First, adding and subtracting the continuous partial currents we can write

$$\Phi_h^b = 2\left(J^{out} + J^{in}\right) \tag{4a}$$

$$\left(\hat{n}\cdot J_h^b\right) = J^{out} - J^{in}. \tag{4b}$$

Using these relationships for vertex i and either setting $\xi_j = 0$ if a face j is internal or on a vacuum boundary or setting $\xi_j = 1$ if a face j is on a reflective boundary, the necessary quantities are

$$\mathbf{a}_j\,\Phi_i^b = 2\,\mathbf{a}_j\left(J_{j,i}^{out}\,[1+\xi_j] + J_{j,i}^{in}\,[1-\xi_j]\right) \tag{5a}$$

$$\left(\mathbf{a}_j\cdot J_i^b\right) = |\mathbf{a}_j|\left(J_{j,i}^{out} - J_{j,i}^{in}\right)[1-\xi_j]. \tag{5b}$$

The upwind definitions are completed by defining the partial currents in these last expressions flowing through face j at the vertex i as

$$J_{j,i}^{in} = \frac{1}{4}\,\Phi_i^{ext} - \frac{1}{2}\,\hat{n}_j\cdot\mathbf{J}_i^{ext} \tag{5c}$$

$$J_{j,i}^{out} = \frac{1}{4}\,\Phi_i + \frac{1}{2}\,\hat{n}_j\cdot\mathbf{J}_i, \tag{5d}$$

where "ext" denotes exterior quantities, that is, the values in the adjacent cell that shares face j with cell k and "across" the face from vertex i. We note that there are other possible ways to define the upwinding although we will not discuss that further here.

If we order the unknowns on the mesh first by the current vector for each vertex on every cell, followed by the scalar flux for each vertex on every cell, we can write the linear system in the (2×2) block form

$$\begin{bmatrix} \mathbf{A}_t & \frac{1}{3}\mathbf{A}_0 \\ -\mathbf{A}_0^T & \mathbf{A}_a \end{bmatrix}\begin{bmatrix} \mathbf{J} \\ \Phi \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ g \end{bmatrix}, \tag{6}$$

where the submatrices $\mathbf{A}_t$ and $\mathbf{A}_a$ are SPD. The system can also be written in the symmetric form

$$\begin{bmatrix} 3\mathbf{A}_t & \mathbf{A}_0 \\ \mathbf{A}_0^T & -\mathbf{A}_a \end{bmatrix}\begin{bmatrix} \mathbf{J} \\ \Phi \end{bmatrix} = \begin{bmatrix} 3\mathbf{f} \\ -g \end{bmatrix}. \tag{7}$$

Finally, we note that it is possible to find a discontinuous discretization for the scalar flux alone if we compute the Schur complement, $\mathbf{S}$, of the (2×2) block linear systems above. The Schur complement is formed by block Gaussian elimination of the row block (which can be viewed as eliminating the currents in favor of the scalar fluxes). The corresponding SPD linear system is

$$\mathbf{S}\Phi = \left[\mathbf{A}_a + \mathbf{A}_0^T(3\mathbf{A}_t)^{-1}\mathbf{A}_0\right]\Phi = g + \left[(3\mathbf{A}_t)^{-1}\mathbf{A}_0\right]f. \tag{8}$$

On tetrahedra this reduced system involves $1/4$ the number of unknowns of the original full P_1 system of equations. This is potentially a tremendous savings for transport acceleration since only the scalar fluxes are needed for that purpose. Solution of the reduced system is a topic we are currently studying.

2.2 Solution Methods

The discontinuous P$_1$ equations, Eq. 6 or Eq. 7, form an sparse, indefinite linear system, $\mathbf{A}x = b$. To be used as part of a DSA scheme, we must solve this system efficiently. For large problems direct methods are infeasible, so we use iterative solution techniques to which we can apply multilevel acceleration methods.

Our solution technique consists of a two–level iteration. An outer Krylov–subspace iterative method is used to solve the discontinuous P$_1$ equations. To compute a solution efficiently, a preconditioner $\mathbf{M}$ is needed that adequately alters the spectrum of the matrix $\mathbf{A}$. Loosely speaking, the preconditioner will be effective if the eigenvalues of the preconditioned matrix $\mathbf{M}^{-1}\mathbf{A}$ are clustered and bounded away from the origin (Campbell, 1996). In our case we have also found that the convergence rate of the outer Krylov iteration improves if the condition number, measured by the ratio of the maximum to minimum singular values of $\mathbf{M}^{-1}\mathbf{A}$, is smaller than that of the original system. However, this is only a rough indicator of preconditioner effectiveness (Warsa, 2000, Nachtigal, 1992). The preconditioner, first presented in Warsa, et al. (2000), has been extended from two–dimensional Cartesian meshes to three–dimensional unstructured tetrahedral meshes. A linear continuous finite element discretization of the diffusion equation is used to precondition the outer Krylov iteration. The inner iteration used to solve the continuous diffusion linear system is the second level of our two–level solution technique.

This preconditioner was shown to be effective over a wide range of problems and is particularly well–suited to solving problems which are optically thick and diffusive. This is fortunate because these happen to be just the kinds of problems for which we would like to solve the P$_1$ equations as part of a transport acceleration algorithm. Use of the continuous diffusion equation is suggested by the observation that discontinuities in the P$_1$ solution disappear and approach the continuous diffusion equation solution as the problem becomes optically thick and diffusive. It is also due in part to the two models being so closely related, a fact we will exploit in deriving our two-level preconditioner. In two dimensions, the preconditioner scaled well with problem size but convergence rates of both the inner and outer iterations degraded for very thin or very highly skewed cells.

At every iteration, the Krylov solver supplies a residual vector r requesting that a vector $z = \mathbf{M}^{-1}r$ be returned to the solver. In our case, this is computed implicitly. That is, we "solve" $\mathbf{M}z = r$ without explicitly forming or inverting a matrix $\mathbf{M}$. This will made clearer if one examines our two–level approach as displayed in Algorithm 1. The algorithm consists of three distinct steps, giving it the character of a two–stage multigrid V–cycle. The continuous diffusion equation step, denoted by $\mathbf{C}^{-1}$, resides at the lowest level and is preceded and followed by damped pre– and

Algorithm 1 Two–Level Preconditioner

```
$z \leftarrow 0$
$s \leftarrow r - \mathbf{A}z$
$z \leftarrow z + \omega\widetilde{\mathbf{A}}^{-1}s$
$s \leftarrow r - \mathbf{A}z$
$z \leftarrow z + \mathbf{C}^{-1}s$
$s \leftarrow r - \mathbf{A}z$
$z \leftarrow z + \omega\widetilde{\mathbf{A}}^{-1}s$
```

post–smoothing iterations, where $\widetilde{\mathbf{A}}$ is some simple approximation to $\mathbf{A}$. It is this entire series of computations that is implicitly represented by the *operator* $\mathbf{M}^{-1}$.

The continuous diffusion *operator*, denoted by $\mathbf{C}^{-1}$ in Algorithm 1, must be computed implicitly as well. The continuous diffusion equation is discretized with unknowns on the mesh vertices. This linear system is usually of lower dimension ($\mathbb{R}^{N_v}$, with N_v being the number of vertices in the mesh) than that of the discontinuous P_1 equations ($\mathbb{R}^{16N_c}$, where N_c is the number of cells in the mesh). The computation of $\mathbf{C}^{-1}$ therefore consists of three steps: a projection, a matrix inversion, and an interpolation, written symbolically as $\mathbf{C}^{-1} = \mathbf{Q}\mathbf{D}^{-1}\mathbf{P}$. The matrix $\mathbf{P}$ projects from from $\mathbb{R}^{16N_c}$ onto $\mathbb{R}^{N_v}$ and the matrix $\mathbf{Q}$ interpolates back again.

The projections and interpolation are also computed implicitly. The projections are computed as a source term for each cell–vertex centered diffusion equation by summing an appropriate combination of the discontinuous scalar flux and current residuals from all the cells surrounding a vertex. This is discussed in more detail below. The interpolations simply assign the same continuous diffusion equation solution to all the discontinuous scalar fluxes surrounding a particular vertex; the currents are left unchanged.

The matrix $\mathbf{D}^{-1}$ represents the inverse of the continuous diffusion equations. The solution is also computed approximately using preconditioned conjugate gradient (PCG) iterations and simple diagonal preconditioning.

The matrix $\widetilde{\mathbf{A}}$ is either the block–diagonal matrix extracted from the discontinuous P_1 matrix on a cell–by–cell basis, represented by $\mathbf{B}$, or it is the identity $\mathbf{I}$. The former case is a block–Jacobi iteration and the latter is a Richardson iteration. In the block–Jacobi iteration, each block is a (16×16) matrix representing the coupling between all the unknowns (scalar fluxes and currents) on a cell. The block–Jacobi smoothing iterations are evaluated by sweeping through the mesh one cell at a time, which could potentially be very efficient in a parallel implementation. The overall preconditioner with Richardson smoothing is simpler but less effective than with block–Jacobi smoothing. In this paper we consider only $\widetilde{\mathbf{A}} = \mathbf{B}$.

Manipulation of the P_1 equations leads to a source term that represents the "correct" projection operator. We start by assuming the discontinuous unknowns in P_1 equations are continuous at the vertices. We then write the balance equations and moment equations (in vector form) for the four vertices on some cell k. The right hand side of the P_1 equations is set to a "residual" vector of the discontinuous operator; referring to Algorithm 1 this vector is $s = r - \mathbf{A}z$, where z is the updated vector from the first block Jacobi iteration. The four moment equations are added together to find an expression for the average current vector on a cell, noting that the area vectors of a cell sum to zero and that the outwardly directed area vectors for a face shared by two cells are the negative of one another. The expression for the average current appears in the four balance equations on a cell. Inserting that expression into the balance equation for vertex j on cell k gives

$$\frac{\mathbf{a}_j}{27\sigma_{t,k}V_k}\cdot\left(\sum_{i=1}^{4}\mathbf{a}_i\,\phi_i\right)+\frac{\sigma_{a,k}V_k}{20}\left(2\phi_j+\sum_{\substack{i=1\\ i\neq j}}^{4}\phi_i\right)=s_{j,k}-\frac{\mathbf{a}_j}{3\sigma_{t,k}V_k}\cdot\left(\sum_{i=1}^{4}\mathbf{s}_{i,k}\right). \tag{9}$$

This is the continuous diffusion discretization with the right hand side being the projection of the discontinuous residual vector s, where $s_{j,k}$ is the residual in the scalar flux at vertex j in cell k and $\mathbf{s}_{j,k}$ is the residual in the currents. The continuous unknowns, ϕ_n, are given the global ordering of the mesh vertices. Every mesh vertex is shared by an arbitrary number of cells and Eq. 9 is computed for the corresponding local vertex j on each of those cells. The equations are summed over the surrounding cells, each one contributing to the coefficients of the continuous diffusion matrix, $\mathbf{D}$, for the row corresponding to that global vertex. The projection operation in an implementation follows from this summation.

We must use GMRES(m) if either the system or the preconditioner is not symmetric. With our definition of the projection operator and the simple interpolation method we are using, the preconditioner is not symmetric because $\mathbf{Q} \neq \mathbf{P}^T$. However, if we wish to use MINRES to solve the symmetric form of the P$_1$ equations, we can impose symmetry by *defining* $\mathbf{Q} = \mathbf{P}^T$ and we can ensure that the preconditioner is positive definite (MINRES requires the preconditioner to be SPD) by scaling the problem by some norm of the linear system.

2.3 Diffusion Synthetic Acceleration

We now present a brief description of the diffusion synthetic accelerated iterative transport solution method.

The DFEM discretization of the S$_N$ transport equation uses the same linear trial functions $u_h \in U_h$ as the P$_1$ equations. We assume an angular quadrature $\{\hat{\Omega}_m, w_m\}$ whose weights sum to unity and consider only isotropic scattering. For each quadrature angle$\hat{\Omega}_m$, the transport discretization computes a *discontinuous* value of the angular flux, $\psi_{m,i,j}$, each vertex j on every cell k. At source iteration ℓ, the discrete form of the transport equation can be written as

$$\mathbf{T}_m \psi_m^{\ell+1} = \sigma_s \, \mathbf{S} \phi^\ell + Q_m, \quad \phi^\ell = \sum_m w_m \psi_m^\ell, \tag{10}$$

where Q_m is an (isotropic) inhomogeneous distributed source. With DSA, the source iteration is modified as follows:

$$\psi_m^{\ell+1/2} = \sigma_s \mathbf{T}_m^{-1} \mathbf{S} \phi^\ell + \mathbf{T}_m^{-1} Q_m \tag{11a}$$

$$\phi^{\ell+1/2} = \sum_m w_m \psi_m^{\ell+1/2} \tag{11b}$$

$$\epsilon^{\ell+1/2} = \sigma_s \, \mathbf{X}^T \mathbf{A}^{-1} \mathbf{X} \left(\phi^{\ell+1/2} - \phi^\ell \right) \tag{11c}$$

$$\phi^{\ell+1} = \phi^{\ell+1/2} + \epsilon^{\ell+1/2}. \tag{11d}$$

The operator $\mathbf{A}^{-1}$ represents the solution to the P$_1$ equations. Note that the scalar flux transport residuals only contribute a source to the balance equations of the P$_1$ equations (the quantity g in Eqs. 6 and 7). Sources for the moment equations are set to zero ($\mathbf{f} = 0$ in Eqs. 6 and 7). Similarly, the transport scalar fluxes are corrected only by the P$_1$ equation solution for the scalar fluxes. The matrix $\mathbf{X}$ symbolizes these “projections” and “interpolations”.

3 Fourier Analysis

In this section we present a three–dimensional Fourier analysis on tetrahedra of the accelerated S$_N$ transport source iteration.

The underlying Fourier ansatz is made on a three–dimensional Cartesian grid, the basic element of which is a box of dimension $\Delta x \times \Delta y \times \Delta z$. The box is divided into some number of tetrahedra whose edges must line up when the basic elements are "translated" in order to "tile" the Cartesian grid with tetrahedra. The minimum number of tetrahedra that satisfy this requirement is six. This basic element is illustrated in Fig. 1.

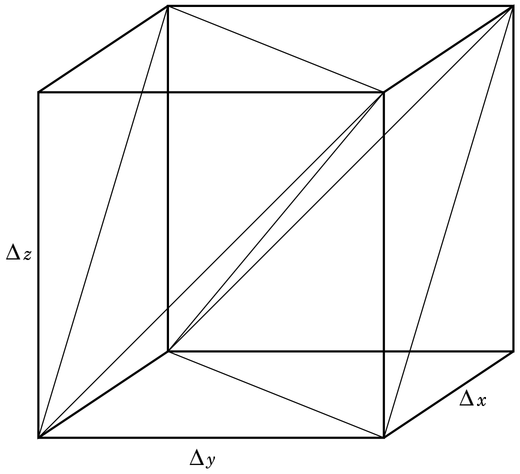

Figure 1: The basic element for the Fourier analysis divided into six tetrahedra of equal volume.

The tetrahedral cells on the basic element are numbered from $k = 1, \ldots, 6$, each of the four vertices are locally ordered within each tetrahedron from $j = 1, \ldots, 4$, and the four quantities at each vertex in every cell in the basic element are ordered as $i = 4(k-1) + j$. A Fourier ansatz is made for the errors in the solution on the basic element. This assumes an exponential representation for the solution of the form

$$\begin{aligned}
\Phi_{j,k}(\mathbf{r}) &= \hat{\Phi}_i e^{\,\mathrm{i}\left(\bar{\Lambda}\cdot\mathbf{r}\right)} \\
J^x_{j,k}(\mathbf{r}) &= \hat{J}^x_i e^{\,\mathrm{i}\left(\bar{\Lambda}\cdot\mathbf{r}\right)} \\
J^y_{j,k}(\mathbf{r}) &= \hat{J}^y_i e^{\,\mathrm{i}\left(\bar{\Lambda}\cdot\mathbf{r}\right)} \\
J^z_{j,k}(\mathbf{r}) &= \hat{J}^z_i e^{\,\mathrm{i}\left(\bar{\Lambda}\cdot\mathbf{r}\right)}
\end{aligned}$$

where $\bar{\Lambda} = \left[\lambda_x, \lambda_y, \lambda_z\right]^T$ is the vector of Fourier wave numbers. The terms for cells in the basic element that have faces on the boundary and which exist outside the basic element (the "external" terms Φ^{ext}_k and $\mathbf{J}^{ext}_k$ in Eqs. 5) are defined in terms of quantities interior to the basic element by "translating" the appropriate interior quantities by the width of the basic element through the Fourier ansatz.

Using the symbolic algebra program MAPLE, Eqs. 3, together with the upwinding equations, Eqs. 5, are written for all cells and all vertices on the basic element. The Fourier ansatz is inserted into these equations resulting in a (96 × 96) linear system. The matrix for the system of equations, denoted by $\widetilde{\mathbf{A}}$, is computed in terms of the basic element thickness Δx, Δy, and Δz, the Fourier wave numbers λ_x, λ_y and λ_z, and *constant* material properties c (scattering ratio) and σ_t (total cross section), using expressions generated symbolically by MAPLE.

For every discrete ordinate m, a Fourier ansatz of the form

$$\psi_{m,j,k}(\mathbf{r}) = \hat{\psi}_{m,i} e^{\,\mathrm{i}\left(\bar{\Lambda}\cdot\mathbf{r}\right)}$$

is also made for the errors in the angular fluxes of the transport equation. Using the same global ordering for the six tetrahedra in a basic element as used in the P$_1$ equations, Eq. 10 leads to the (24×24) system

$$\widetilde{\mathbf{F}} = \sum_m w_m \widetilde{\mathbf{T}}_m^{-1} \widetilde{\mathbf{S}}. \tag{12}$$

This matrix is again computed in terms of the Fourier wave number, the basic element thickness, and the material properties, using symbolic expressions from MAPLE. Note that the spectral radius of the unaccelerated source iterations, the maximum eigenvalue of$\widetilde{\mathbf{F}}$, is equal to $c = \sigma_s/\sigma_t$.

Now, the DSA algorithm in Eqs. 11a leads to the (24×24) matrix ($\mathbf{I}$ is the identity)

$$\widetilde{\mathbf{G}} = \left[\widetilde{\mathbf{F}} + \sigma_s \, \widetilde{\mathbf{X}}^T \widetilde{\mathbf{A}}^{-1} \widetilde{\mathbf{X}} \big(\widetilde{\mathbf{F}} - \mathbf{I} \big) \right], \tag{13}$$

whose maximum eigenvalue is the spectral radius of the accelerated transport solution. The matrix $\widetilde{\mathbf{X}}$ is a discrete “projection” matrix of dimensions (96×24).

The Fourier matrix $\widetilde{\mathbf{G}}$ is computed with MAPLE by combining symbolic expressions for $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{F}}$, and $\widetilde{\mathbf{X}}$. It is evaluated for fixed parameters and maximized over all frequencies using a Nelder–Mead simplex algorithm with quadratic surface fitting near suspected maxima. We found this algorithm to be essential in searching for the maximum over the three–dimensional space of wave numbers.

4 NUMERICAL RESULTS

In this section we will investigate the effectiveness and efficiency of the fully consistent DSA (FCDSA) scheme using our solution method for the P$_1$ equations. Numerical results are computed using our implementation code, ATTILAV2, as described in Wareing, et al. (2001). Theoretical predictions of the spectral radius are computed using the results of the Fourier analysis of the previous section.

In the results that follow, Fourier analysis predictions and measurements of the spectral radius will be given for the FCDSA scheme. They will be compared to those of the partially consistent DSA method of Adams and Martin, called the Modified Four Step Method (M4S) (Adams, 1992). In all the computations reported here, we use Algorithm 1 with$\widetilde{\mathbf{A}} = \mathbf{B}$ to solve Eq. 7 using GMRES. Note that same algorithm can be used to accelerate the M4S DSA equations.

ATTILAV2 currently uses a partially consistent DSA scheme developed by Wareing, et al. (1991), which we refer to as the WLA (Wareing, Larsen, and Adams) method. Although we have not performed Fourier analysis of this method, we will report measured values of the spectral radius since WLA DSA is available in the implementation code.

For isotropic scattering, we anticipate the fully consistent method will be stable for all cell widths and cell aspect ratios.

In Table 1 we compare the spectral radius predicted by Fourier analysis to the measured spectral radius using the ATTILAV2 transport code for a scattering ratio $c = 0.9999$ and total cross section $\sigma_t = 3.5$ cm^{-1}. The results are tabulated for representative cell widths (in terms of the size of the basic element “box”) in order of decreasing aspect ratio α. The aspect ratio measure is computed as the three times the ratio of the inscribed to circumscribed spheres. It is less than or equal to one, attaining its maximum for a tetrahedron with edges of equal length and approaching zero as the tetrahedra become more distorted (Liu, 1994). The minimum aspect ratio, α_{min}, of the six tetrahedra in the basic element is listed in the table. The measured spectral radius is computed on a fixed $(8 \times 8 \times 8)$ grid of boxes, each divided into six tetrahedra, for a total of 3072 cells in the problem. The outer problem dimensions are varied to alter the aspect ratio of the tetrahedral cells; for example, to have a basic element of size $(2.0 \text{ cm} \times 1.0 \text{ cm} \times 5.0 \text{ cm})$, the problem domain is $x \in [0, 16 \text{ cm}], y \in [0, 8 \text{ cm}], z \in [0, 40 \text{ cm}]$. Boundary conditions on the bottom, left, and back faces of the problem are reflective, the others are vacuum. Sources are set to zero and the angular fluxes are initialized randomly and the scalar fluxes are normalized to the total scalar flux in the problem after each iteration. We use an S_4 triangular, Chebyshev-Legendre (TCL) quadrature and a relative convergence criterion of 10^{-4} for the inner iterations. We use a tolerance of 10^{-5} for the PCG iterations on the continuous diffusion equations in Algorithm 1. The spectral radius is measured as the ratio of the change in the discrete L2 norm between successive iterations, reported at the end of 100 iterations. The WLA spectral radius is reported after 300 iterations.

The results in Table 1 indicate that the fully consistent DSA method is stable and effective whereas the partially consistent method of Adams and Martin can be unstable when the aspect ratio is small. The measured spectral radius is, as expected, less than the Fourier analysis prediction because there is leakage from the vacuum boundaries. The M4S and WLA methods show a strong dependence on cell aspect ratio and cell thickness. The M4S method becomes unstable as the aspect ratio decreases. This is an unexpected result since the initial verification of the method in one– and two–dimensions did not indicate any instability (Adams, 1992). The spectral radius of the WLA method increases as the aspect ratio decreases. However, we want to emphasize that this is for a scattering ratio very close to unity and in general the method is very effective when the scattering ratio is not so close to one. Nonetheless, in applications any degradation in effectiveness of the WLA DSA scheme is often compensated by its computational efficiency. The fully consistent method remains effective over the range of cell aspect ratios.

Table 1: Fourier analysis prediction and computationally measured spectral radius tabulated as a function of tetrahedral cell aspect ratio.

				FCDSA		M4S		WLA
α_{min}	Δx	Δy	Δz	Measured	Analysis	Measured	Analysis	Measured
0.632	1.0	1.0	1.0	0.2171	0.2215	0.5036	0.5298	0.6542
0.562	2.0	2.0	3.0	0.1808	0.1706	0.5632	0.5851	0.8441
0.487	1.0	1.0	2.0	0.2092	0.2062	0.6138	0.6420	0.7877
0.421	2.0	2.0	5.0	0.1768	0.1714	0.6704	0.7043	0.8975
0.370	2.0	1.0	3.0	0.2044	0.2066	0.9156	0.9586	0.8461
0.327	3.0	1.0	3.0	0.1982	0.2008	1.0315	1.0783	0.8470
0.256	2.0	1.0	5.0	0.2100	0.2123	1.0622	1.1173	0.9002
0.170	2.0	1.0	8.0	0.2024	0.2054	1.1590	1.2254	0.9327
0.116	8.0	1.0	10.0	0.1462	0.1482	1.4320	1.4755	0.9452

The next set of results are shown in Fig. 2. We compare the measured spectral radius and the Fourier analysis for the three DSA schemes as before. In our measurements the scattering ratio is again taken to be $c = 0.9999$. This time, however, the total cross section is varied logarithmically from 2^{-5} cm^{-1} to 2^{10} cm^{-1} and the basic element size is fixed $(2.0$ cm $\times$ 1.0 cm $\times$ 8.0 cm$)$ for which the minimum cell aspect ratio is 0.170. The measured spectral radius was computed on a fixed $(6 \times 6 \times 6)$ grid of boxes, each divided into six tetrahedra, for a total of 1296 cells. The problem domain is fixed at $x \in [0, 12$ cm$], y \in [0, 6$ cm$], z \in [0, 48$ cm$]$. The boundary conditions again consist of three reflective faces and three vacuum faces. Again we use the S_4 TCL quadrature and the same inner iteration convergence criteria for the discontinuous P_1 equations.

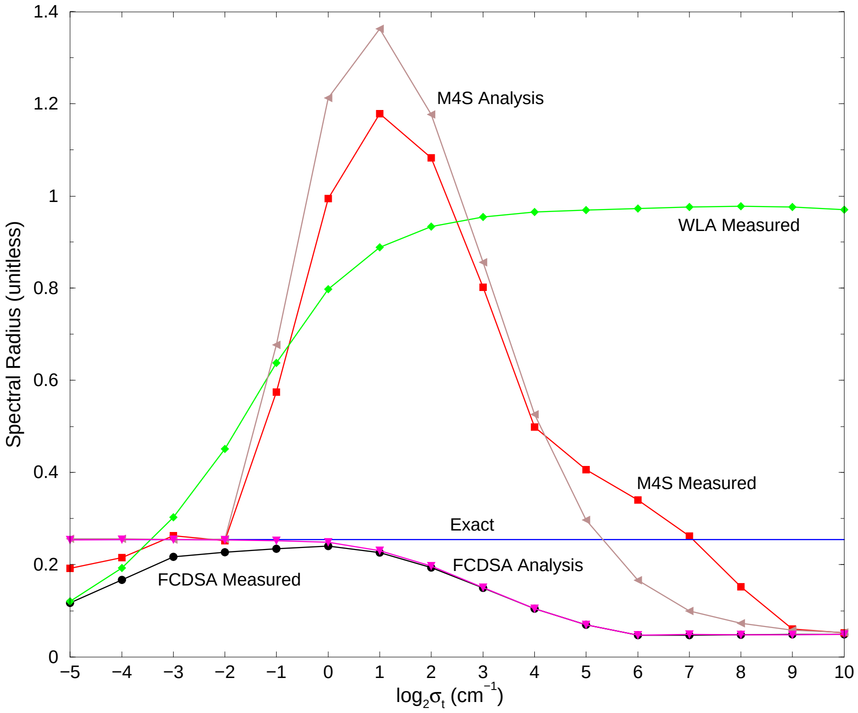

Figure 2: Fourier analysis and computationally measured spectral radius shown as a function of total cross section.

We expect that the spectral radius will approach zero for optically thick cells and that it should approach the value $\tilde{\rho}_0 = 0.2542$ in the limit of vanishing cell thickness for the TCL S_4 quadrature as used in the transport code (Larsen, 1982, Adams, 1993). This value of $\tilde{\rho}_0$ is shown in the figure for comparison. For thin cells, the Fourier analysis approaches the exact result. The measured results drop off from the exact result because of leakage as the problem becomes extremely thin. The fully consistent scheme is stable and effective for all cell optical thickness. The M4S method becomes unstable for intermediate optical thickness. While both the FCDSA and M4S methods become increasingly effective as the problem becomes thick and diffusive, the effectiveness of the WLA method degrades with increasing optical thickness.

The last set of results compares the computational effort needed to compute an S_N solution with the FCDSA method to that of the WLA DSA scheme. For this problem, we solve a one–group, steady state, oil well logging tool problem, a 139.7 cm tall half cylinder of radius 60 cm, modeled with an unstructured mesh of 43,012 cells (Wareing, 1996). There are two He-3 detectors and a unit source of neutrons inside the problem. The minimum aspect ratio for the mesh is 0.1234 while the maximum is 0.9996. The problem contains the isotropically scattering materials whose properties are listed in Table. 2.

Table 2: Neutron cross sections for the oil well logging tool problem.

	Cross sections (cm^{-1})	
Material	Total	Scattering
Limestone	$8.79672 \cdot 10^{-1}$	$8.70516 \cdot 10^{-1}$
Iron	$1.16776 \cdot 10^{0}$	$9.66125 \cdot 10^{-1}$
Water	$3.13459 \cdot 10^{0}$	$3.11519 \cdot 10^{0}$
He-3	$4.94621 \cdot 10^{-1}$	$1.00243 \cdot 10^{-4}$

The number of floating point operations (FLOP) needed to compute the S_N solution to a relative convergence criteria of 10^{-5} in the scalar fluxes on an SGI Origin 2000 processor are shown in Table. 3 for a range of TCL quadrature orders N. The FLOP count is a measure of computational effort not affected by memory access issues and is independent of data layout or other system resource use.

The highly scattering, diffusive water–containing regions of the problem brings the unaccelerated spectral radius to approximately 0.9916. This is the measured spectral radius for the S_4 quadrature which used $266.4 \cdot 10^9$ FLOP to converge in 2234 iterations. The FLOP count is roughly proportional to the solution time; for comparison, the unaccelerated solution was computed in 215 CPU minutes. Noting that this is only a one group problem (the cross sections correspond to the lowest neutron energy group in a 47 group library), it is clear that DSA is necessary for practical applications.

Because of the low minimum cell aspect ratio, the M4S DSA method is unstable. For example, after the third iteration with the S_4 quadrature the spectral radius is 1.5136 and the method never recovers. This observation is independent of quadrature order.

The table shows that the WLA method is very efficient with each DSA step taking a very small fraction of the total solution time. in to the FCDSA scheme. In this case, the DSA algorithm takes the majority of the computation time. However, because it is so effective in reducing the spectral radius – FCDSA converged in 13 iterations and WLA in 102 – the overall computation time can be less than the more efficient WLA method for high enough quadrature order.

5 CONCLUSIONS

We have found that our fully consistent DSA scheme for DFEM discretizations of the S_N equations based on an analogous DFEM discretization of the P_1 equations is stable and very effective over a wide range of cell shapes, dimensions and optical thicknesses. For problems with a low aspect ratio, we found that the partially consistent M4S DSA scheme can be unstable on 3D unstructured grids while the effectiveness of the WLA DSA method degrades for optically thick, diffusive problems.

We expected that the increased complexity of the fully consistent DSA scheme could make it impractical for many problems. Still, it could be more computationally efficient than the partially consistent WLA DSA method under certain circumstances. For example, some problems may require a high quadrature order because they contain streaming regions while also having regions

Table 3: Floating point operation counts (in billions) for the oil well logging tool problem for various S_N quadrature orders N. The total needed to compute the S_N solution is tabulated along with the counts spent in just the DSA algorithm (percent of total is in parentheses).

	FCDSA		WLA	
N	Total	DSA	Total	DSA
4	111.4	110.0 (98.7%)	13.7	1.683 (12.3%)
8	115.4	110.6 (95.8%)	41.5	1.684 (4.06%)
12	120.6	110.6 (91.7%)	85.0	1.684 (1.98%)
16	128.2	111.1 (86.7%)	144.5	1.684 (1.17%)
20	137.1	111.1 (81.0%)	219.8	1.684 (0.77%)
24	148.0	111.1 (75.1%)	310.9	1.684 (0.54%)

in the problem that are diffusive. Another application where the fully consistent method could be more efficient than the WLA scheme is in certain radiative transfer problems of interest that have a scattering ratio very close to one.

So far we have implemented our methods in serial codes only. Our conclusions could change when we extend our method to parallel platforms.

We plan to pursue methods for solving the reduced Schur complement system for use in DSA applications. Solving the reduced system could make the fully consistent DSA method more competitive. There are applications other than DSA that require the full discontinuous P_1 solution. In that case an efficient solution of the reduced system could be used as part of a very effective preconditioner for the full system. We are planning to explore this possibility.

ACKNOWLEDGEMENTS

The authors would like to thank Michele Benzi for sharing with us his expertise in numerical linear algebra. This work was performed under the auspices of the U.S. Department of Energy at the Los Alamos National Laboratory.

REFERENCES

Adams, M. L., Martin, W. R., 1992. Diffusion synthetic acceleration of discontinuous finite element transport iterations. *Nucl. Sci. and Engr.* **111**, 145.

Adams, M. L., Wareing, T. A., 1993. Diffusion-synthetic acceleration given anisotropic scattering, general quadratures, and multidimensions. *Trans. of the Am. Nucl. Soc.* **68**, 203.

Alcouffe, R. E., 1977. Diffusion synthetic acceleration methods for diamond–differenced discrete–ordinates equations. *Nucl. Sci. and Engr.* **64**, 344.

Campbell, S. L., Ipsen, I. C. F., Kelley, C. T., Meyer, C. D., 1996. Gmres and the minimal polynomial. *BIT* **36**, 664.

Larsen, E. W., 1982. Unconditionally stable diffusion–synthetic acceleration methods for slab geometry discrete ordinates equations. Part I: Theory. *Nucl. Sci. and Engr.* **82**, 47.

Liu, A., Joe, B., 1994. Relationship between tetrahedron shape measures. *BIT* **34**, 268.

Morel, J. E., Dendy, J. E., Wareing, T. A., 1993. Diffusion–accelerated solution of the two–dimensional S_n equations with bilinear–discontinuous differencing. *Nucl. Sci. and Engr.* **115**, 304.

Nachtigal, N. M., Reddy, S. C., Trefethen, L. N., 1992. How fast are nonsymmetric matrix iterations? *SIAM J. Matrix Anal. and Apps.* **13**, 778.

Wareing, T. A., Larsen, E. W., Adams, M. L., 1991. Diffusion accelerated discontinuous finite element schemes for the sn equations in slab and x–y geometries. In: *Proc. International Topical Meeting on Adavances in Mathematics, Computations, Reactor Physics*, Volume 3, Pittsburgh, Pennsylvannia, p. 11.1 2.1.

Wareing, T. A., McGhee, J. M., Morel, J. E., 1996. Attila: A three–dimensional, unstructured tetrahedral mesh discrete ordinates transport code. *Trans. of the Am. Nucl. Soc.* **75**, 146–147.

Wareing, T. A., McGhee, J. M., Morel, J. E., Pautz, S. D., 2001. Discontinuous finite element S_n methods on 3–D unstructured grids. *Nucl. Sci. and Engr.* (to appear, July 2001).

Warsa, J. S., Wareing, T. A., Morel, J. E., 2000. Solution of the discontinuous P_1 equations in two–dimensional cartesian geometry with two–level preconditioning. *SIAM J. Sci. Comp.* (submitted).

Zienkiewicz, O. C., Taylor, R. L., 1994. *The Finite Element Method*, Volume 1. McGraw–Hill, London, Fourth Edition.